**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №**4

**по дисциплине «Построение и анализ алгоритмов»**

**Тема: Алгоритм Кнута-Морриса-Пратта**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8304 |  | Птухов Д.А. |
| Преподаватель |  | Размочаева Н.В. |

Санкт-Петербург

2020

**Вариант 1.**

## Цель работы.

Построение и анализ алгоритма Кнута-Морриса-Пратта на основе решения задачи о нахождении циклического сдвига строки.

## Основные теоретические положения.

Заданы две строки A (|A|∣ ≤5000000) и B (∣*B*∣≤5000000). Определить, является ли А циклическим сдвигом В (это значит, что А и В имеют одинаковую длину и А состоит из суффикса В, склеенного с префиксом В). Например, defabc является циклическим сдвигом abcdef.

## Описание алгоритма.

Для определения величины сдвига было использовано только последнее значение префикс функции для уже расширенной строки. Далее при помощи проверки в цикле было определено являются ли оставшиеся суффикс первой строки и префикс второй идентичными. Если да, то итоговое значение равно последнему значению префикс функции, иначе первая строка не может быть получена из второй при помощи сдвига. Оценка сложности – пусть размер первой строки – n, размер второй – m, тогда префиксная функция работает за O(n) + считывание второй строки O(m) + проверка идентичности – O(n). Ответ – O(2n + m).

**Описание функций.**

Для реализации вышеописанного алгоритма была реализована функция shift, принимающая входной и выходной поток. Данная функция осуществляет посимвольное считывание 2-ой строки (для экономии памяти) и параллельное вычисление значения префикс функции дял конкатенации двух входных строк. Далее осуществляется проверка на то – являются ли строки идентичными, если да то возвращается 0 иначе осуществляется вышеописанная проверка префикса и суффикса первой и второй строки соответственно.

**Вывод промежуточной информации.**

Во время основной части работы алгоритма происходит вывод промежуточной информации, а именно, значения префикс функции и проверка идентичности префикса и суффикса первой и второй строки соответственно.

**Тестирование.**

Таблица 1 – Результаты тестирование

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| defabc  abcdef | 3 |
| defabz  abcdef | -1 |
| Baa  aaB | 2 |
| GggHgg  gggggg | -1 |

# Вывод.

В ходе работы был построен и анализирован алгоритм Форда-Фалкерсона на основе решения задачи о нахождении циклического сдвига. Код программы представлен в приложении А.

**ПРИЛОЖЕНИЕ А.  
ИСХОДНЫЙ КОД**

#include <iostream>

#include <string>

#include <vector>

void prefixx(std::string& s, std::vector<size\_t>& pi)

{

pi[0] = 0;

for (size\_t i = 1; i < s.size(); ++i)

{

size\_t j = pi[i - 1];

while (j > 0 && s[i] != s[j])

j = pi[j - 1];

if (s[j] == s[i])

++j;

pi[i] = j;

}

}

int shift()

{

std::string s;

std::cin >> s;

size\_t start\_size = s.size();

std::vector<size\_t> pi(s.size());

prefixx(s, pi);

char c;

size\_t j = 0;

size\_t text\_ind = 0;

bool is\_same = true;

while (std::cin >> c)

{

if (s[text\_ind] != c)

is\_same = false;

while (j > 0 && s[j] != c)

j = pi[j - 1];

if (s[j] == c)

++j;

pi[start\_size + j] = j;

++text\_ind;

}

if (is\_same)

return 0;

if (text\_ind != start\_size - 1)

return -1;

size\_t ind = pi[pi.size() - 1];

for (int i = ind; i < start\_size; ++i)

{

if (s[i] != s[i + start\_size - ind])

return -1;

}

return start\_size - ind;

}

int main()

{

std::cout << shift();

return 0;

}